SECTION 5
INSTALLATION INSTRUCTIONS FOR SOLARIS 2.3

5.1 DETAILED PROCEDURES FOR SOLARIS 2.3. The hardware, when delivered by Sun Microsystems, will not have the operating system loaded. Therefore, it is assumed that the loading of all DSRS-related COTS support software will begin by loading Solaris Version 2.3.

All commands given in this document are for the **C Shell.** To invoke the C shell environment, enter the command:

```
/bin/csh
```

The SunOS will provide the special username, **root**, for use during system administrative activities. When a user logs in with this username, they will become the most privileged user on the system, the *superuser.* A superuser will have permission to run critical system administration programs and edit sensitive files (privileges that are denied to regular users). A user can become superuser by either: logging in at the console using **root** as the username, or typing **su** from a shell where he/she is logged in under the regular user name.

If the C2 security environment has been properly installed, the root account may only log into the SunOS at the console terminal. To perform superuser functions from anywhere other than the console terminal, it is necessary to use the **su** command (to become the superuser).

NOTE: When the steps in this document instruct you to log into the root account, you will either need to **su** to superuser or execute the installation from the console terminal.

NOTE: When the steps in this document reference the ORACLE_SID of dsrsdb, you may use the value which has been defined for your site. The ORACLE_SID value selected must then be used consistently throughout the installation procedure. The ORACLE_SID value cannot be **dsrs**.

5.1.1 Preparation. Before installation, the following must be determined:

a. **Who is to be the Sun System Administrator**? An individual at each site must be assigned as the site's Sun System Administrator and adequately trained in UNIX and Sun-specific System Administrator functions.

b. **Where will the COTS software be loaded**? The directories in which the COTS software products will be installed will need to be determined before installation. Vendor's product documentation should be consulted for any required directory structures which must be put in place.

c. **What are the passwords for the root and ORACLE accounts**? The site's Sun System Administrator will be responsible for assigning and maintaining all system level passwords.

d. **Where will the DSRS be loaded**? The directory in which the DSRS will be installed will need to be determined before installation. The examples in this document use **/usr/DSRS** as the directory where the DSRS will be loaded.

e. **Which tape drive will be used**? The tape drive that will be used to install the system must be accessible. See Section 5.6.1 for information on the tape controllers and their devices.

f. **What will be the DSRS owner account?** The username for the DSRS owner account will need to be determined. The username for the DSRS owner account provided in the examples for this document is **dsrstest.**

g. **What are the IP Address and Hostname?** The IP address and hostname for the local site will need to be determined.

5.1.2 Setup. Prior to the installation of DSRS-related COTS software, a "clean machine" environment will be established. "Clean machine" is defined as a microcomputer of the Sun 4 architecture as delivered by the vendor, with no software or operating system installed.

5.2 INSTALLATION OF COTS. This section contains information on the procedures employed to install and test the required DSRS-related COTS software products. It is recommended that the software products described below be installed in the presented order; however, this is not mandatory.

5.2.1 Installation Procedures for Run-Time Environment COTS Support Software Packages. The following procedures will be utilized in the installation and testing of those DSRS-related COTS support software packages that are to be installed.

It is pertinent to mention here that the majority of the Sun software packages can be loaded by exercising options during the installation of the SunOS Operating System. Likewise, the majority of the ORACLE products can be loaded during installation of the ORACLE7 Server (RDBMS) software.

5.2.1.1 Installation Procedures for Solaris. Follow the loading and testing instructions contained in the *Solaris 2.3 System Configuration and Installation Guide* manual. Installation and configuration information is found in the section, "Installing Solaris on a Server", which contains information necessary for installing Solaris Version 2.3 on different types of workstations and servers, as well as troubleshooting procedures.

The installation procedures will allow the installation to be performed in a series of steps. When performing an installation it is recommended that the installer resize the root and swap partitions on the disk to allocate the swap space to be 2 1/2 times the amount of memory on the system, and to remove the h partition on drives with space limitations. Systems that have a limited amount of disk space might prefer to perform a custom installation to allow the selection of only the Solaris software categories which will be applicable to the site.

If the site will be running the Domain Name System (DNS), then the DSRS host must be either a Network Information Service (NIS) server or a client of a NIS server.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation;

See Section 5.10.4 for information on debugging Solaris problems.

The site's Sun System Administrator will be responsible for creating the necessary directory structure to accommodate installation of the COTS support software and DSRS.

Prior to installing ORACLE it is necessary to reconfigure the kernel to include the ORACLE parameters. These parameters with ORACLE's recommended values are as follows:

```
set shmsys:shminfo_shmmax=8388608
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=100
set shmsys:shminfo_shmseg=10
set semsys:seminfo_semmns=250
set semsys:seminfo_semmni=70
set semsys:seminfo_semmsl=20
```

After the parameters have been added to the **/etc/system** file the system may be rebooted to incorporate the parameters.

5.2.1.2 **Installation Procedures for OpenWindows.** Installation instructions for OpenWindows Version 3.3 are contained in the *Solaris 2.3 System Configuration and Installation Guide.*

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.1.3 **Installation Procedures for ICS OSF Motif.** All available information regarding the loading of this package is found in the ICS OSF/Motif™ 1.2.2 *Installation and Release Notes* and manual.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

After ICS OSF Motif has been installed it is required that the following commands be performed.

```
# cp /opt/ICS/Motif/usr/lib/libXm.so.2.2 /opt/ICS/Motif/lib/libXm.so.2
# cp /opt/ICS/Motif/usr/lib/libMrm.so.2.2 /opt/ICS/Motif/usr/lib/libMrm.so.2
```

NOTE: Solaris 2.4 includes Motif runtime libraries that allow tools that were developed using Motif to run.

To use the Solaris Motif runtime libraries comment the following line in /dsrscm/config/xinitrc:

```
/opt/ICS/Motif/usr/bin/X11/mwm &
```

and add the following:

```
/usr/openwin/bin/olwm &
```

The ICS OSF Motif software is not required for sites using the Solaris Motif runtime libraries.

5.2.1.4 **Installation Procedures for ORACLE7 Server (RDBMS).** Information regarding the procedures to load the ORACLE7 Server are found in the *ORACLE7 for Sun SPARC Solaris 2.3 Installation and User's Guide Release 7.1.3*. Chapters 5 and 6 prepare the user for the installation. Chapters 7 and 8 contain the procedures for installing ORACLE. Refer to Table 5-I for recommended installation responses.

The ORACLE installation requires an account for the ORACLE owner. Refer to Section 5.6.2 for detailed steps to add user accounts. It is recommended that the default shell for the ORACLE account be the C-Shell.

Before starting the ORACLE installation process it is necessary to have the following environment variables defined, as stated in Chapter 5 of the *ORACLE7 for Sun SPARC Solaris 2.3 Installation and User's Guide Release 7.1.3.*

ORACLE_HOME - defines the directory where the ORACLE software will be installed.

ORACLE_TERM - defines the terminal definition resource file to be used during the installation.

ORACLE_SID - defines the system identifier of the database you will create with the installer.

ORACLE SQL*Loader is automatically installed during the ORACLE7 Server (RDBMS) installation.

It is recommended that all ORACLE products be installed in one ORACLE installation session. Review all of the ORACLE product installation sections before beginning the installation described in this section.

During the ORACLE installation, the user will be prompted with a series of questions to define environment variables. To install ORACLE7 Server (RDBMS) the user must select ORACLE7 Server (RDBMS) from the Available Products menu.

The following error message may appear in the $ORACLE_HOME/orainst/install.log file during the installation process and can be ignored:

```
ranlib:  warning:  filename; no symbol table
```

Upon completion of the ORACLE7 Server installation, log in as superuser and execute the file $ORACLE_HOME/orainst/root.sh.

It is advisable to have the document cited in the first paragraph of this section available, as much peripheral information is found in other chapters.

a. Vendor-supplied testing procedures or instructions:

   Unable to locate in available documentation.

b. Vendor-supplied error messages:

   ORACLE7 Server Messages and Codes Manual Release 7.1.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation;

See Section 5.10.3 for information on debugging ORACLE problems.

Table 5-I. ORACLE Installation Information

Screen Title	Prompt	Recommended Input	Recommended Response
ORACLE_HOME	Enter the pathname for your ORACLE_HOME directory:	/usr/oracle7	(OK)
ORACLE_DOC	Enter the pathname for your ORACLE_DOC directory:	/usr/oracle7	(OK)
/cdrom/oracle/orainst/ README.FIRST			(OK)
Skip README	Would you like to skip the displaying of this README file in the future?		(Yes)
ORACLE Owner	Enter the name of the ORACLE owner:	oracle	(OK)
Install Actions		Install/Upgrade/Patch Software Only	(Select)
Online Document Load		For no products.	(Select)
NLS	Select the native language to be installed:	American/English	(Select)
Relink All Executables	Would you like to relink Oracle product executables?		(Yes)
Information	The installation log will be written to /usr/oracle7/ orainst/install.log		(OK)
Solaris 2 Documentation	Do you want the port-specific documentation installed?		(No)
Product Documentation Library CD-ROM Install	Select the desired CD-ROM product documentation installation	For no products.	(Select)

Screen Title	Prompt	Recommended Input	Recommended Response
Available Products		ORACLE7 Server (RDBMS) 7.1.3.0.0 PL/SQL V2 2.1.3.0.0 SQL*Plus 3.1.3.4.1 SQL*Net TCP(V1) 1.2.7.8.1 Pro*C 1.6 1.6.4.1.0[1]	(Install...)
Install TCP/IP Protocol Adapter (V2)?	Would you like to install TCP/IP Protocol Adapter (V2)?		(No)
Install DECNet Protocol Adapter (V2)?	Would you like to install DECNet Protocol Adapter (V2)?		(No)
Install OSI4 Protocol Adapter (V2)?	Would you like to install OSI4 Protocol Adapter (V2 BETA)?		(No)
Install Async Protocol Adapter (V2)?	Would you like to install Async Protocol Adapter (V2)?		(No)
DBA Group	Please select the group which should be able to act as the DBA of the database.	dba	(Select)
ORACLE_SID	Enter your ORACLE_SID:	oradbs	(OK)

[1] Install in addition to above list for Development Environment COTS Support Software.

5.2.1.5 Installation Procedures for ORACLE SQL*Plus. Information regarding the procedures to load SQL*Plus are found in Chapter 7 of the *ORACLE7 for Sun SPARC Solaris 2.3 Installation and User's Guide Release 7.1.3*. Additionally, it is advisable to have *SQL*Plus User's Guide and Reference Version 3.1* available for detailed information about the capabilities and intricacies of the software package.

To install SQL*Plus, the user must select SQL*Plus 3.1.3.4.1 from the Available Products menu during the ORACLE installation procedure described in Section 5.2.1.4.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Appendix A - Copy Command Error Messages, *SQL*Plus User's Guide and Reference Version 3.1;*

ORACLE7 Server Messages and Codes Manual Release 7.1.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.1.6 **Installation Procedures for SQL*Net for TCP/IP.** Information regarding the procedures to load SQL*Net are found in Chapters 6, 7 and 9 of the *ORACLE7 for Sun SPARC Solaris 2.3 Installation and User's Guide Release 7.1.3*. Additionally, it is advisable to have the *SQL*Net User's Guide Version 1.2* available for detailed information on how to use SQL*Net TCP/IP.

To install SQL*Net TCP/IP the user must select SQL*Net TCP/IP (v1) 1.2 from the Available Products menu during the ORACLE installation procedure described in Section 5.2.1.4.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

ORACLE7 Server Messages and Codes Manual Release 7.1.

*SQL*Net TCP/IP User's Guide Version 1.2.*

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.1.7 **Installation Procedures for ORACLE PL/SQL.** Information regarding the procedures to load ORACLE PL/SQL are found in Chapter 2 of the *ORACLE for Sun SPARC Solaris 2.3 Installation and User's Guide Release 7.1.3*. Additionally, it is advisable to have the *PL/SQL User's Guide and Reference Version 2.0* available for detailed information on Oracle's procedural language extension to SQL.

To install ORACLE PL/SQL the user must select PL/SQL 2.1.3.0.0 from the Available Products menu during the ORACLE installation procedure described in Section 5.2.1.4.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

ORACLE7 Server Messages and Codes Manual Release 7.1.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.1.8 **<u>Installation Procedures for Minerva MSQL RDBMS.</u>** The Minerva MSQL RDBMS installation cannot be performed until the DSRS is installed. Information on the procedures to install the Minerva MSQL RDBMS are found in the *README*.

a. The user must be a "supervisor" and must be in the **/usr** directory.

b. If the **/usr/local** directory does not currently exist, create it.

```
# mkdir /usr/local
```

c. Create the **/usr/local/Minerva** directory.

```
# mkdir /usr/local/Minerva
```

d. Change directory to **/usr/local/Minerva**.

```
# cd /usr/local/Minerva
```

e. Extract the files from the **/dsrscm/utils/msql-sol23.tar**.

```
# tar -xvf /dsrscm/utils/msql-sol23.tar
```

f. Start the **msql** daemon. The following line will start the **msql daemon** during the boot process when inserted into a file located in the **/etc/rc2.d** directory.

```
# /usr/local/Minerva/bin/msqld &
```

The following line will be displayed and can be ignored:

```
Couldn’t open ACL file: No such file or directory
```

g. Databases may be created with the following command, using a unique database name, in place of ***dbname.*** A database must be created for each DSRS X/Motif user and the database name ***dbname*** must be the same as the Unix user name.

```
# /usr/local/Minerva/bin/msqladmin create dbname
```

h. For each msql database created with the **msqladmin** script the following lines must be added to the **/usr/local/Minerva/msql.acl** file. The unique database name ***dbname*** will be the same as the DSRS X/Motif user name.

database = ***dbname***
read = ***dbname***
write = ***dbname***
host = *
access = local, remote

i. Vendor-supplied testing procedures or instructions:

README

j. Vendor-supplied error messages:

Unable to locate in available documentation.

k. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.1.9 Installation Procedures for freeWAIS-0.5. Information on the procedures to load freeWAIS-0.5 are found in the documentation obtained in the FTP download of freeWAIS-0.5. See Section 3.2 for information on obtaining freeWAIS-0.5.

a. Vendor-supplied testing procedures or instructions:

Sample test scripts are supplied in the wais-test directory.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.2 Installation Procedures for Development Environment COTS Support Software Packages. The following procedures will be utilized in the installation and testing of the DSRS-related COTS support software packages for the development environment (to be installed by SRP personnel).

Depending on site requirements, one or more of these products may be loaded by the SRP software installation team.

5.2.2.1 Installation Procedures for SPARCworks-Pro-C. Information on the procedures to be followed for loading Sun SPARCworks-Pro-C are found in the SunPro *SPARCworks Product Evaluation Guide*.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.2.2 **Installation Procedures for Sun SPARCompiler C++.** Information on the procedures to load SPARCompiler C++ are found in the *SunSoft Workshop Developers Products.*

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.2.3 **Installation Procedures for XVT DSC++.** Information on the procedures to load XVT DSC++ are found in the *Installing XVT Development Solution for C++ for Motif (SPARC Platforms)*.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.2.4 **Installation Procedures for XVT-Graphical Extensions.** Information on the procedures to be followed for loading XVT for C are found in the *Installing XVT Development Solution for C for Motif (SPARC Platforms).*

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.2.2.5 **Installation Procedures for ORACLE Pro*C.** Information on the procedures to load Pro*C are found in Chapter 2 of the *ORACLE7 for Sun SPARC Solaris 2.3 Installation and User's Guide Release 7.1.3.*

Prior to loading Pro*C, it is mandatory that the appropriate versions of ORACLE7 Server (RDBMS) and the Sun SPARC C compiler have been installed.

To install Pro*C, the user must select Pro*C 1.6.4.0.0 from the Available Products menu during the ORACLE installation procedure described in Section 5.2.1.4.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Appendix D - Error Messages, *Programmer's Guide to the ORACLE Precompilers Version 1.5.*

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Chapter 7 - Handling Runtime Errors, *Programmer's Guide to the ORACLE Precompilers Version 1.5.*

5.2.2.6 **Installation Procedures for SPARCworks Professional C++.** Information on the procedures to load SPARCworks Professional C++ are found in the *SunSoft Workshop Developers Products*.

a. Vendor-supplied testing procedures or instructions:

Unable to locate in available documentation.

b. Vendor-supplied error messages:

Unable to locate in available documentation.

c. Vendor-supplied debugging/troubleshooting procedures or instructions:

Unable to locate in available documentation.

5.3 INSTALLATION OF DSRS. The following information describes the steps necessary to load and install the DSRS software (after the appropriate COTS support software environment has been created).

5.3.1 Setup. The following steps are necessary before DSRS installation:

a. Create an account for the DSRS owner. Refer to Section 5.6.2 for detailed steps to add user accounts. This account name cannot be **dsrs**.

b. Create the directory where the distribution tape will be loaded. This directory is the DSRS owner home directory pathname.

mkdir /usr/DSRS

If the directory has already been created, the following error will occur and can be ignored.

mkdir: /usr/DSRS file exists

c. Change ownership of the above directory to belong to the DSRS owner account.

chown dsrstest /usr/DSRS

d. Create the **dsrsadm** groups. The **gid** field must be unique for each group on the node. The **gid** of the **dsrs** group must be the same **gid** used when creating the DSRS owner account. Refer to Section 5.6.3 for detailed steps to create groups.

e. Make the DSRS owner account a member of the **dsrsadm** and **dba** groups. Refer to Section 5.6.3 for detailed steps to assign the DSRS owner account to the **dsrsadm** and **dba** groups.

5.3.2 Installation. This section explains how to read the DSRS software from the tape media to the target directory, and execute the script file to create the database. The examples shown below use the environment variables: ORACLE_SID = dsrsdb; ORACLE database directory location = /usr/oracle7/dbs; account performing the installation = dsrstest; tape device = /dev/rmt/0. For sites using a value different than those documented above, replace the values specified in this document with those currently being used at the installation site. Refer to

Section 5.6.1.1 to determine the tape devices available on your system. This installation defines the ORACLE_SID and the ORACLE database name to be the same.

a. Log into the DSRS owner account created in the setup section. After you have logged in, you are automatically in the directory where you will read the DSRS distribution tape.

```
Login: dsrstest
```

b. Insert the DSRS distribution tape into the tape drive.

c. Read the tape into the target directory. To read the tape, type:

```
tar xvpf /dev/rmt/0 > tapelist.log
```

The tape device may not be **/dev/rmt/0**; therefore, the correct tape drive device information will need to be substituted for **/dev/rmt/0** in the above command. Refer to Section 5.6.1.1 for detailed information on determining the tape devices available. The screen output will be captured to the file tapelist.log.

d. The **.cshrc** and **.login** files must be created in the DSRS owner account home directory to correctly define the necessary environment variables. Update both files with your site-specific configuration.

The examples given here contain only the basic information needed for DSRS operation in each of these files.

(1) Copy the **.cshrc** file to the DSRS owner home directory. The correct directory location for ORACLE_HOME and the correct value for ORACLE_SID will need to be entered for the local site.

```
cp /usr/DSRS/config/cshrc ~dsrstest/.cshrc
chmod 700 ~dsrstest/.cshrc
```

Verify that the following entries are correct.

```
setenv ORACLE_HOME /usr/oracle7
set path = ($PATH $ORACLE_HOME/bin)
setenv ORACLE_SID dsrsdb
```

(2) Copy the **.login** file to the DSRS owner home directory.

```
cp /usr/DSRS/config/login ~dsrstest/.login
chmod 700 ~dsrstest/.login
```

Verify that the following entries are correct.

```
set ORAENV_ASK=NO; source $ORACLE_HOME/bin/coraenv
unset ORAENV_ASK
```

e. Using the editor of your choice, edit the file **/usr/DSRS/testdb/init.two** and replace the third line which reads, **"db_name = dsrs"**, with the correct value of the ORACLE_SID which will be created during the installation.

```
db_name = dsrsdb
```

f. Using the editor of your choice, edit the file **/usr/DSRS/testdb/init.dsrs** and replace the third line which reads, **"db_name = dsrs",** with the correct value of the ORACLE_SID which will be created during the installation.

```
db_name = dsrsdb
```

g. Log into the root account using the **su** command. It will be necessary to insert the password when prompted.

```
su -
```

h. Create a symbolic link to the **/usr/DSRS** directory.

```
ln -s /usr/DSRS /dsrscm
```

To verify that the link has correct ownership and protections as listed below:

```
ls -al /dsrscm
```

file protections - lrwxrwxrwx
owner - root

i. Change file protections for all files in the **/dsrscm/testdb** directory for access by the ORACLE account:

```
chmod -R 777 /dsrscm/testdb/*
```

j. Log out of the root process.

```
exit
```

5.3.3 **<u>Create DSRS Database.</u>** The following information describes the steps necessary to create the DSRS database. Sites which aleady have a DSRS database do not need to create a new DSRS database.

a. Log into the ORACLE account, enter the password at the password prompt:

```
su oracle
```

b. Execute the **setup.csh** script file to define all environment variables for the database installation process. The script file **setup.csh** is located in the directory structure defined for the dsrstest home directory pathname.

```
source /dsrscm/bin/setup.csh
```

If the following error is encountered:

```
source:  not found
```

then execute the two following commands to define the C Shell environment and to execute the **setup.csh** script file.

```
/bin/csh
source /dsrscm/bin/setup.csh
```

c. Create the directories where the ORACLE database files will be installed. Below is an example using the database directory which has already been created during the ORACLE installation and a directory which is on another disk device. Any valid directory may be specified for the location of the database files. Refer to the *System Administration Manual* for information pertaining to disk devices and filesystems.

```
mkdir  /usr/oracle7/dbs
mkdir /usr2/indexes
```

If the following error is encountered:

```
mkdir:  /usr/oracle7/dbs:  No such file or directory
```

then execute the two following commands to create the **/usr/oracle** directory and the **/usr/oracle/dbs** directory.

```
mkdir /usr/oracle7
mkdir /usr/oracle7/dbs
```

d. Using the editor of your choice, edit the file **$ORACLE_HOME/dbs/init.ora** and replace the line that reads, **"db_name = DEFAULT"**, with the correct value of the ORACLE_SID, which will be created during the installation.

```
db_name = dsrsdb
```

e. The **dsrs_install** script file is provided for installation and will create a new database, instance and add all database objects required and loads the test database information. It is highly recommended that the **dsrs_install** script be used. The following command can be used to remove the database created by the ORACLE install script, if desired.

```
/dsrscm/testdb/remove_database ORACLE_SID db_file_path index_file_path
```

The three DSRS tablespaces will be created in the directory defined when the **dsrs_install** procedure is initiated. See Table 5-I for the disk space required for the tablespaces. The DSRSINDXS tablespace must be created on a disk device other than the device where the DSRS and DSRS_ROLLBK are to be created. Refer to the *System Administration Manual* for information on disk devices and partitions.

Execute the **dsrs_install** script file which will:

(1) Check for existing database files and ORACLE_SIDs that might already exist on the system.

(2) Copy the file $ORACLE_HOME/dbs/init.ora into $ORACLE_HOME/dbs /init$ORACLE_SID.ora.

(3) Execute $DSRS_TESTDB/create_database to create the database.

(4) Execute $DSRS_TESTDB/create_user to create the necessary ORACLE user accounts for the DSRS.

(5) Execute $DSRS_TESTDB/create_tablespace to create the three DSRS tablespaces defined in Table 5-I.

The three DSRS tablespaces will be created in the directories defined when the **dsrs_install** procedure is initiated.

(6) Execute $DSRS_TESTDB/sys_rollbk to create a second rollback segment in the system tablespace.

(7) The ORACLE database is then shutdown and restarted using the parameter file $DSRS_TESTDB/init.two.

(8) Execute $DSRS_TESTDB/add_rollbk.segs to create the DSRS rollback segments.

(9) The ORACLE database is then shutdown and restarted using the parameter file $DSRS_TESTDB/init.dsrs.

(10) Copy the file $DSRS_TESTDB/init.dsrs into $ORACLE_HOME/dbs/init $ORACLE_SID.ora.

(11) Execute $DSRS_TESTDB/add_table_defs to create the DSRS database tables.

(12) Execute $DSRS_TESTDB/add_sequence to create the guest sequence.

(13) Execute $DSRS_TESTDB/reset_db to import the database.

(14) Execute $DSRS_TESTDB/add_indexes to create the indexes.

(15) Execute $DSRS_TESTDB/grant_access.

A unique ORACLE_SID will need to be entered on the command line; this value must be the value defined as the db_name in steps f and g, **dsrsdb**, the directory structure where the ORACLE database files will be created, **/usr/oracle7/dbs**, and the directory structure where the ORACLE index file will be created, **/usr2/indexes**. Refer to Appendix A for the information that will be generated and displayed on the screen when the following command is executed.

```
$DSRS_TESTDB/dsrs_install dsrsdb /usr/oracle7/dbs /usr2/indexes
```

f. The program will terminate and the operating system prompt will appear.

g. Log out of the ORACLE account.

h. Log out of the DSRS account.

Refer to Sections 5.10.3 and 5.10.4 for information on debugging ORACLE and Solaris problems that may occur.

5.3.4 Enable ORACLE Auditing. This section will need to be performed for all DSRS databases. Refer to Appendix J for all steps required for creating and using the ORACLE database audit capabilities.

5.4 POST DSRS INSTALLATION. The following steps are necessary to complete the DSRS installation. All SunOS accounts which belong to users who are either librarians or supervisors of the DSRS and will be using the **dsrs_import** utility must be assigned membership to the **dsrsadm** group. The **dsrsadm** group members will be allowed access to the protected DSRS executables.

a. Log into the root account or **su** to become superuser.

b. The file protection must be updated to point to the dsrsadm group id created in Section 5.3.1, step d, for the following files. Enter the following commands to update the group protection for the DSRS librarian and supervisor executables:

```
chgrp dsrsadm /dsrscm/testdb/*
```

c. The file protection must be updated to disallow world users access to the following files:

```
chmod 770 /dsrscm/testdb/*
```

d. Copy the dsrs resource file **/dsrscm/config/dsrs** to the **/usr/openwin/lib/app-defaults** directory, and the Motif app-defaults directory.

```
cp /dsrscm/config/dsrs /usr/openwin/lib/app-defaults/dsrs
cp /dsrscm/config/dsrs /opt/ICS/Motif/usr/lib/apps-defaults/dsrs
```

e. Create a symbolic link to the **/dsrscm/dsrs_utilities/captive** file.

```
ln -s /dsrscm/dsrs_utilities/captive /bin/captive
```

To verify that the link has correct ownership and protections.

```
ls -al /bin/captive
```

```
file protections - lrwxrwxrwx
owner - root
```

f. Create the **/var/adm/dsrs** directory.

```
mkdir /var/adm/dsrs
```

g. Log out of the root account.

5.4.1 Logon Shell Scripts. For each user account on the DSRS, you must define the necessary environment variables when initiating the account. This can be accomplished by creating **.login**, **.cshrc, .xinitrc, .mwmrc** and **.Xdefaults** files. These files are provided with the installation and may be copied to every DSRS user's home directory. Examples of the **.cshrc** and **.login** files may also be found in Appendix B.

```
cp /dsrscm/config/login  ~username/.login
cp /dsrscm/config/cshrc  ~username/.cshrc
cp /dsrscm/config/dsrs ~username/.Xdefaults
cp /dsrscm/config/xinitrc ~username/.xinitrc
cp /dsrscm/config/mwmrc ~username/.mwmrc
chown username ~username/.login
chown username ~username/.cshrc
chown username ~username/.Xdefaults
chown username ~username/.xinitrc
chown username ~username/.mwmrc
```

NOTE: The above files may need to be updated with your site-specific configuration and ORACLE_SID, MOTIF_HOME, and DISPLAY environment variable for your IP address.

5.4.2 Configuring the DSRS Servers. The DSRS Servers must be properly configured to allow for client connections. This section will step you through editing the necessary files, modifying the database, creating a new database export file, logging into the DSRS servers and updating the database information.

a. Log into the root account.

b. Use the **admintool** to update the services database to the following services:

```
iop_srv   1600/tcp      # Interop Server for DSRS
pcdsrs    1604/tcp      # pc_server for DSRS
libdsrs    1610/tcp     # lib_server for DSRS
```

c. Using the editor of your choice, edit the file **/etc/inetd.conf** and append the line shown below:

```
iop_srv stream tcp nowait  root  /dsrscm/bin/dsrs_server.sol23 dsrs_server
pcdsrs stream tcp nowait root /dsrscm/bin/pc_server.sol23 pc_server
libdsrs stream tcp nowait root /dsrscm/bin/lib_server.sol23 lib_server
```

d. Signal the internet daemon so it will re-read its configuration file, using the following command:

```
kill -HUP pid
```

e. Use the **admintool** to update the hosts database to add entries for each remote system with which you interoperate. Insert the IP addresses and host names to define the remote servers and host.

f. Using the editor of your choice, edit the file **/dsrscm/bin/dsrs_server.sol23** and insert the correct ORACLE_SID for your local site.

g. Using the editor of your choice, edit the file **/dsrscm/bin/pc_server.sol23** and insert the correct ORACLE_SID for your local site

h. The file protections for the DSRS server files will be 755.

```
# chmod 755 /dsrscm/bin/pc_server.sol23
# chmod 755 /dsrscm/bin/dsrs_server.sol23
# chmod 755 /dsrscm/bin/lib_server.sol23
```

i. Using the editor of your choice, edit the file **/var/opt/oracle/oratab** and insert the line below, using the correct ORACLE_SID and ORACLE_HOME of the database created in Section 5.3.2:

```
dsrsdb:/usr/oracle7:Y
```

j. Log out of the root account.

k. Log into the DSRS owner account.

l. Rename the current **/dsrscm/testdb/testdb.exp** to **/dsrscm/testdb/testdb.exp_del**:

```
mv /dsrscm/testdb/testdb.exp  /dsrscm/testdb/testdb.exp_del
```

m. The database must be properly configured to allow for remote extracts and ftp extracts. This section will step you through updating the database and creating a new test database import file.

   (1) Execute the commands defined in the *Librarian Manual for the DSRS* to update the **ip address** and **hostname** for the local site, using the correct ip address and hostname.

n. Export the database information into a new **/dsrscm/testdb/testdb.exp** which will be used for all system tests.

```
$ORACLE_HOME/bin/exp rcl/brodie  file=/dsrscm/testdb/testdb.exp     \
        grants = y
```

o. Logout of the DSRS owner account.

5.4.3 **Install Minerva MSQL RDBMS.** Refer to Section 5.2.1.8 for detailed steps to install and configure the Minerva MSQL RDBMS.

5.4.4 **Configure Oracle SQL*Net for TCP/IP.** The following steps are necessary to complete the Oracle SQL*Net configuration for server machines.

a. Log into the root account or **su** to become superuser.

b. The file permissions must be made as follows:

```
chown root $ORACLE_HOME/bin/orasrv

chmod 4555 $ORACLE_HOME/bin/orasrv
```

c. Use the **admintool** to update the services database and add the service shown below:

```
orasrv 1525/tcp #SQL*Net orasrv
```

d. Signal the internet daemon, **inetd**, so it will re-read its configuration file, using the following command:

```
kill -HUP pid
```

e. To start orasrv during the boot process, insert the following line into the **/etc/rc2** file:

```
su - oracle -c /usr/oracle7/bin/orasrv
```

f. Verify that SQL*Net TCP/IP is configured correctly by using the checkTCP utility. The commands below will run all of the checkTCP system tests:

```
cd $ORACLE_HOME/tcp/install
checkTCP -a
```

5.5 DATA UPDATE PROCEDURES. The procedures described in the following sections are provided for sites that need to upgrade their DSRS V6.0 or DSRS V5.2 databases to the DSRS V6.05 database format. Section 5.5.1 provides the steps necessary to upgrade a DSRS V6.0 database to a V6.05 database, and Section 5.5.2 will provide the steps necessary to upgrade a DSRS V5.2 database to a V6.05 database.

Sites updating their DSRS V5.0 databases to a DSRS V6.05 database must first convert the DSRS V5.0 database to a DSRS V5.1, then convert the DSRS V5.1 database to a DSRS V5.2.

5.5.1 DSRS V6.0 to V6.05 Database Changes. The script files described in the following sections are provided for sites that want to upgrade their DSRS V6.0 databases to the DSRS V6.05 database format.

The database conversion will be performed in the following five phases. At the completion of each phase a database export file will be created to allow the user to reset the database and then execute the next phase.

Steps Executed	Section
Execute update60to605_1 which will spool current table data and create temporary conversion tables.	5.5.1.1
Execute update60to605_2 which will spool ra_id and ra_version into ra_id.lis and version.lis.	5.5.1.2
Execute update60to605_3 which will create the /dsrscm/key files.	5.5.1.3

Steps Executed	Section
Execute update60to605_4 which will drop necessary tables, recreate the inactive_classification and ra_use tables, and load spooled data into the tables.	5.5.1.4
Execute update60to605_5_sol23 which will create the /dsrscm/wais directories and execute $DSRS_BASE/lib/index_keywords to generate the keyword database.	5.5.1.5

The database conversion will include the following changes:

a. A keyword file will be created for each RA that was assigned to a domain in the DSRS V6.0 database. The keyword file will be created in the /dsrscm/key directory and will consist of all unique facet terms used for classifying the RA in the DSRS V6.0 database. The keyword filename will be in the form of raid_version.key.

b. The following tables will be deleted from the database: D[*domain_id*]_CLASSIFICATION, DESCRIPTOR, and FACET.

c. A column will be added to the RA_USE table. This column will be RA_USE.VISIBLE and values will be set to 1 for True.

d. The facet and facet_term columns will be deleted from the INACTIVE_CLASSIFICATION table.

e. Entries for each RA Key file will be entered into the RA_FILE table.

NOTE: The above executables will take a couple of hours to complete. The total time necessary will depend on the number of Key files to be created and if Oracle Audit capability is enabled. On an IPX it took over 2 hours for the process to complete 1565 RAs without Oracle Audit enabled.

NOTE: SQL*Loader may create the following two types of output files:

***.log** As SQL*Loader executes, it creates a file where it stores information about the loading process.

***.bad** As SQL*Loader executes, it creates a file where it places records that had formatting errors or caused ORACLE errors. Duplicate records during the SQL*Loader execution will generate *bad.files.

5.5.1.1 **<u>Execute update60to605_1.</u>** The steps below require the **/dsrscm** directory to point to the new DSRS 6.05 directory structure. If this has not been properly defined, please configure it before executing the steps below.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.0 database.

```
setenv ORACLE_SID dsrsdb
```

c. Create a directory from which to execute the database conversion. The **update60to605_1** script file will create files during phase 1 of the database update.

```
mkdir /dsrscm/testdb/v60tov605/upgrade
```

d. Change to the directory created in the above step.

```
cd /dsrscm/testdb/v60tov605/upgrade
```

e. Execute the file **$DSRS_TESTDB/v60tov605/update60to605_1** which will:

(1) Check that the **/dsrscm/key** directory exists.
(2) Check that the user has write access to the **/dsrscm/key** directory.
(3) Check that the user has write access to the current directory.
(4) Check that the tablespaces have enough free space.
(5) Export the RCL tables into the **begin_1.exp** file.
(6) Spool necessary DSRS V6.0 table data into the following files: **ra_use.dat, inactive_class.dat, inactive_ra.dat** and **drop_classification_tables.sql**
(7) Create the **INACTIVE_RA**, **CLASS** and **CLASS_RA** tables.
(8) For all domains, select the distinct values from the **D[*domain_id*]_CLASSIFICATION** table and insert them into the **CLASS** table.
(9) Spool the distinct **ra_id, ra_version**, and **facet_term** values from the **CLASS** table into the file **class.dat**.
(10) Spool the distinct **ra_id** and **ra_version** values from the **CLASS** table into the file **class_ra.dat**.
(11) Drop the **CLASS** table.
(12) Recreate the **CLASS** table with a primary key.
(13) Export the RCL tables into the **update_1.exp** file.

The following command will perform the above steps.

```
$DSRS_TESTDB/v60tov605/update60to605_1
```

f. Verify that the following files contain DSRS V6.0 table data without any Oracle error messages, and that begin_1.exp and update_1.exp files exist in the current directory. The files are: class.dat, class_ra.dat, inactive_class.dat, inactive_ra.dat, and ra_use.dat.

5.5.1.2 **<u>Execute update60to605_2.</u>** The steps below require the **/dsrscm** directory to point to the new DSRS 6.05 directory structure. If this has not been properly defined please configure it before executing the steps below.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.0 database.

```
setenv ORACLE_SID dsrsdb
```

c. Change to the directory created in step c of Section 5.5.1.1.

```
cd /dsrscm/testdb/v60tov605/upgrade
```

d. Execute the file **$DSRS_TESTDB/v60tov605/update60to605_2** which will:

(1) Check that the user has write access to the current directory.
(2) Load data into the **CLASS** table.
(3) Spool the distinct **ra_id** and **ra_version** from the **CLASS** table into the file **class_ra.dat.**
(4) Load the data from step 4 into the **CLASS_RA** table.
(5) Load data from **inactive_ra.dat** into the **INACTIVE_RA** table.
(6) Spool the **ra_version** from **CLASS_RA** into the file **version.lis** and spool the **ra_id** from **CLASS_RA** into the file **ra_id.lis.**
(7) Spool the new key file information for each **ra_id** from the **CLASS_RA** table into the file **insert_file.dat.**
(8) Split the **ra_id.lis** and **version.lis** files into 250 lines pieces.
(9) Export the RCL tables into the update_2.exp file.

The following command will perform the above steps:

```
$DSRS_TESTDB/v60tov605/update60to605_2
```

5.5.1.3 **<u>Execute update60to605_3</u>.** The steps below require the **/dsrscm** directory to point to the new DSRS 6.05 directory structure. If this has not been properly defined, please configure it before executing the steps below.

a. Log into the DSRS owner account.

b. Define ORACLE_SID environment variable for the DSRS V6.0 database.

```
setenv ORACLE_SID dsrsdb
```

c. Change to the directory created in step c of Section 5.5.1.1.

```
cd /dsrscm/testdb/v60tov605/upgrade
```

d. Execute the file **$DSRS_TESTDB/v60to605/update60to605_3** which will:

(1) Check that the user has write access to the current directory.
(2) For each RA in the **CLASS** table spool the distinct **facet_terms** into a **/dsrscm/key/raid_version.key** file.
(3) Export the RCL tables into the update_3.exp file.

The following command will perform the above steps.

```
$DSRS_TESTDB/v60tov605/update60to605_3
```

5.5.1.4 **Execute update60to605_4.** The steps below require the **/dsrscm** directory to point to the new DSRS6.05 directory structure. If this has not been properly defined, please configure it before executing the steps below.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.0 database.

```
setenv ORACLE_SID dsrsdb
```

c. Change to the directory created in step c of Section 5.5.1.1.

```
cd /dsrscm/testdb/v60tov605/upgrade
```

d. Execute the file **$DSRS_TESTDB/v60tov605/update60to605_4** which will:

(1) Check that the user has write access to the current directory.
(2) Drop all **D*[domain_id]*_CLASSIFICATION** tables.
(3) Drop **CLASS, DESCRIPTION, INACTIVE_RA,** and **RA_USE** tables.
(4) Recreate the **INACTIVE_CLASSIFICATION** and **RA_USE** tables with the new table definitions.
(5) Add the primary key for the **INACTIVE_CLASSIFICATION** table.
(6) Load data into the **INACTIVE_CLASSIFICATION** table.
(7) Load modified data into the **RA_USE** table.
(8) Load the key file data into the **RA_FILE** table.
(9) Grant access to all tables.
(10) Export the RCL tables into the update_4.exp file.

The following command will perform the above steps:

```
$DSRS_TESTDB/v60tov605/update60to605_4
```

5.5.1.5 **Execute update60to605_5_sol23.** The steps below require the /dsrscm directory to point to the new DSRS 6.05 directory structure. If this has not been properly defined, please configure it before executing the steps below.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.0 database.

```
setenv ORACLE_SID dsrsdb
```

c. Change to the directory created in step c of Section 5.5.1.1.

```
cd /dsrscm/testdb/v60tov605/upgrade
```

d. Execute the file **$DSRS_TESTDB/v60tov605/update60to605_5** which will:

(1) Check that the **/dsrscm/wais** directory exists.
(2) Check that the user has write access to the **/dsrscm/wais** directory.
(3) Check that the user has write access to the current directory.
(4) Create the **/dsrscm/wais/domain_*domain_id*** directories for each domain.
(5) Execute **/dsrscm/bin/index_keywords**.**sol23** for all **RAs** in each domain for each Key file.
(6) Export the RCL tables into the update_5.exp file.

The following command will perform the above steps:

```
$DSRS_TESTDB/v60tov605/update60to605_5_sol23
```

5.5.2 **DSRS V5.2 to V6.05 Database Changes.** The script files described in the following paragraphs are provided for sites that want to upgrade their DSRS V5.2 databases to the DSRS V6.05 database format.

The database conversion will be performed in the following six phases. At the completion of each phase a database export file will be created to allow the user to reset the database and then execute the next phase.

Steps Executed	Section
Create spool script for all RA file information.	5.5.2.1
Export the DSRS V5.2 Database.	5.5.2.2
Execute update52to605_1 which will drop all current tables, import DSRS V5.2 table data, and spool all table data.	5.5.2.3

Execute update52to605_2 which will delete single quotes, drop all V5.2 tables, create all DSRS V6.05 tables, and use SQL*Loader to load all table data into the DSRS V6.05 tables.	5.5.2.4
Execute update52to605_3 which will insert values into the USER_OPTION and DOMAIN tables, create DSRS user accounts, spool RA_ID information, and create the /dsrscm/key files.	5.5.2.5
Execute update52to605_4_sol23 which will insert the key file to RA assignments, drop the temporary tables, create the /dsrscm/wais directories, and execute $DSRS_BASE/lib/index_keywords to generate the keyword database.	5.5.2.6

The database conversion will include the following changes:

a. The keyword file will be created in the /dsrscm/key directory and will consist of all unique facet terms used for classifying the RA in the DSRS V5.2 database. The keyword filename will be in the form of raid_version.key.

b. All environment variables will be replaced with the actual directory structure in the RA_FILE table.

c. The user ids will have the letter ‘U’ concatenated to the beginning of the user id.

d. The new fields for RA.TOTAL_SIZE and RA_FILE.FILE_SIZE will be set to 0.

e. All single quotes will be removed from the database entries.

f. A column will be added to the RA_USE table. This column will be RA_USE. VISIBLE and values will be set to 1 for True.

g. The key files will be inserted into the RA_FILE table.

NOTE: The above executables will take a couple of hours to complete. The total time necessary will depend on the number of Key files to be created and if the ORACLE Audit capability is enabled. On an IPX it took over 2 hours for the process to complete 1565 RAs without ORACLE Audit enabled.

NOTE: SQL*Loader may create the following two types of output files.

__*.log__ As SQL*Loader executes, it creates a file where it stores information about the loading process.

***.bad** As SQL*Loader executes, it creates a file where it places records that had formatting errors or caused ORACLE errors. Duplicate records during the SQL*Loader execution will generate *.bad files.

5.5.2.1 **Spool RA File Information.** To perform a DSRS V5.2 database conversion it is necessary to update the file **/dsrscm/testdb/v52tov605/spool_ra_files.sql** and include the environment variable translation for each environment variable for each database table where RA files exist. For example, the test database delivered with the DSRS V5.2 system had the environment variables **DSRS_DOC** and **DSRS_CODE**, which were defined to be **/dsrscm/test_rsc_doc** and **/dsrscm/test_rsc_code**. Each environment variable will require the following six blocks of code to be added to the **/dsrscm/testdb/v52tov605/spool_ra_files.sql** file. The blocks of code are similar, and it is very important that they are edited correctly. The examples below use the example of **DSRS_DOC**.

```
select rsc_id||'|'||rsc_id_version||'|'|| -
document_name||'|'||document_version||'|'|| -
2||'|'||type_of_document||'|'|| -
'/dsrscm/test_rsc_doc/ '||location_filename||'|'|| -
file_type||'|'||0||'|'||sysdate||'|' -
from rcl.document
where location_pathname =  'DSRS_DOC'

/

select rsc_id||'|'||version||'|'|| -
'Source'||'|'||1||'|'||1||'|'||' -
''||'|'||'/dsrscm/test_rsc_doc/'||location_filename||'|'|| -
0||'|'||0||'|'||timestamp||'| -
from rcl.rsc
where location_pathname =  'DSRS_DOC'

/

select rsc_id||'|'||version||'|'|| -
'Source' ||'|'||1||'|'||1||'|'|| -
''||'|'||'/dsrscm/test_rsc_doc/ '||location_filename||'|'|| -
0||'|'||0||'|'||timestamp||'|'-
from rcl.inactive_rsc
where location_pathname =  'DSRS_DOC'

/

select rsc_id||'|'||version||'|'|| -
'Abstract' ||'|'||1||'|'||0||'|'|| -
''||'|'||'/dsrscm/test_rsc_doc/'||abstract_location_filename||'|'|| -
```

```
0||'|'||0||'|'||timestamp||'|'-
from rcl.rsc
where abstract_location_pathname =  'DSRS _DOC'

/

select rsc_id||'|'||version||'|'|| -
'Abstract' ||'|'||1||'|'||0||'|'|| -
''||'|'||'/dsrscm/test_rsc_doc/'||abstract_location_filename||'|'|| -
0||'|'||0||'|'||timestamp||'|'-
from rcl.inactive_rsc
where abstract_location_pathname =  'DSRS_DOC'

/

select rsc_id||'|'||rsc_id_version||'|'|| -
'Problem Report' ||'|'||1||'|'||3||'|'|| -
''||'|'|| '/dsrscm/test_rsc_doc/' ||location_filename||'|'|| -
0||'|'||0||'|'||sysdate||'|' -
from rcl.problem_rpt
where location_pathname =  'DSRS_DOC'

/

exit;
```

Each block spools the following table columns from each of the necessary tables to create the new DSRS6.05 RA_FILE table data.

The columns are as follows and are separated by a ‘|’; the ‘||’ are concatenation symbols.

RSC_ID
RSC_ID Version
Document Name
Document Version
Type (0=Abstract, 1=Source, 2=Document, 3=Problem Report)
Description (where applicable)
Location pathname || location_filename
File Type (0=ASCII, 1=Binary)
Timestamp (sysdate where applicable)

The actual directory structure of location_pathname, as defined by each environment variable, will need to be properly defined in each block of code.

The new blocks must be placed directly before the **exit**. The six blocks may be cut and pasted into the file as many times as necessary to avoid having to type them in.

The script file may be tested with the following command while in an active SQL session.

```
SQL>@/dsrscm/testdb/v52tov605/spool_ra_files
```

Testing of this script will allow you to view any possible errors.

5.5.2.2 **Export V5.2 Database.** The DSRS V5.2 database must be exported into a dump file that will be converted and loaded into a DSRS V6.05 database.

a. Define the ORACLE environment variables and the ORACLE_SID for the DSRS V5.2 database to your process.

```
setenv ORACLE_HOME /usr/oracle
setenv ORACLE_SID dsrsdb
source $ORACLE_HOME/bin/coraenv
```

b. Execute the following command to create a DSRS V5.2 dump file:

```
$ORACLE_HOME/bin/exp rcl/brodie file=dsrsv52.tab grants=n
```

c. Define the ORACLE7 environment variables and the ORACLE_SID for the DSRS V6.0 database to your process.

```
setenv ORACLE_HOME /usr/oracle7
setenv ORACLE_SID dsrsdb
source $ORACLE_HOME/bin/coraenv
```

5.5.2.3 **Execute update52to605_1** The steps below require the **/dsrscm** directory to point to the new DSRS 6.05 directory structure. If this has not been properly defined, please configure it before executing the steps below. The steps below will perform phase 1 of the database conversion process.

a. Log into the DSRS owner account.
b. Define the ORACLE SID environment variable for the DSRS V6.05 database.

```
setenv ORACLE_SID dsrsdb
```

c. Create a directory from which to execute the database conversion. The update52to605_1 script file will create files during phase 1 of the database update.

```
mkdir /dsrscm/testdb/v52tov605/upgrade
```

d. Change to the directory created in the above step.

```
cd /dsrscm/testdb/v52tov605/upgrade
```

e. Execute the file **$DSRS_TESTDB/v52tov605/update52to605_1** which will:
   (1) Check that the user has write access to the current directory.
   (2) Check that the tablespaces have enough free space.
   (3) Drop all tables owned by RCL.
   (4) Import the DSRS V5.2 database export file that was created in Section 5.5.2.2.
   (5) Spool the DSRS V5.2 table data into text files.
   (6) Export the RCL tables into the update_1.exp file.

f. Verify that the following files contain DSRS V5.2 table data without any ORACLE error messages, and that the update_1.exp file exists in the current directory. The files are: abstract_request.d, group_id.d, ra_file.d, add_users.sql, group_person_xref.d, ra_metric.d, asset_request.d, group_ra_xref.d, ra_use.d, class.dat, group_site_xref.d, related_ra.d, class_ra.dat, inact_rel_ra.d, relation.d, d0_ra.d, inactive_class.d, search_metrics.d, drop_all.sql, insert_file.dat, site.d, metric.d, person.d, and ra.d.

The following command will perform the above steps; the second value of the command line is the directory location and filename of the DSRS V5.2 export file created in Section 5.5.2.2.

```
$DSRS_TESTDB/v52tov605/update52to605_1 /dsrscm/dsrsv52.tab
```

5.5.2.4 **Execute update52to605_2.** The steps below will perform phase 2 of the database conversion process. The update52to605_1 script must have executed successfully before beginning this section.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.05 database.

```
setenv ORACLE_SID dsrsdb
```

c. Change to the directory created in Section 5.5.2.3, step c.

```
cd /dsrscm/testdb/v52tov605/upgrade
```

d. Execute the file **$DSRS_TESTDB/v52tov605/update52to605_2** which will:

   (1) Execute convert_files to remove all single quotes from the table data.
   (2) Drop all tables owned by RCL.
   (3) Create the DSRS V6.05 table definitions.
   (4) Create the temporary database conversion tables.
   (5) Add unique and primary keys to the DSRS V6.05 tables.
   (5) Use SQL*Loader to load all V5.2 table data into DSRS V6.05 tables.
   (6) Export the RCL tables into the update_2.exp file.

e. Check the log files created by SQL*Loader. Verify that the update_2.exp file exists in the current directory.

The following command will perform the above steps:

$DSRS_TESTDB/v52tov605/update52to605_2

5.5.2.5 **Execute update52to605_3.** The steps below will perform phase 3 of the database conversion process. The update52to605_2 script must have executed successfully before beginning this section.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.05 database.

setenv ORACLE_SID dsrsdb

c. Change to the directory created in Section 5.5.2.3, step c.

cd /dsrscm/testdb/v52tov605/upgrade

d. Execute the file **$DSRS_TESTDB/v52tov605/update52to605_3** which will:

(1) Check that the **/dsrscm/key** directory exists.
(2) Check that the user has write access to the **/dsrscm/key** directory.
(3) Check that the user has write access to the current table.
(4) Insert values into the USER_OPTION table.
(5) Create the DSRS user accounts.
(6) Grant access to all tables.
(7) Spool RA and RA_VERSION files.
(8) Split the **ra_id.lis** and **version.lis** files into 250 lines pieces.
(9) Execute create_files to spool the distinct **facet_terms** into a **/dsrscm/key/raid_version.key** file.
(10) Export the RCL tables into the update_3.exp file.

e. Check that files were successfully created in the /dsrscm/key directory.

Verify that the update_3.exp file exists in the current directory.

The following command will perform the above steps.

$DSRS_TESTDB/v52tov605/update52to605_3

5.5.2.6 **Execute update52to605_4_sol23.** The steps below will perform phase 4 of the database conversion process. The update52to605_3 script must have executed successfully before beginning this section.

a. Log into the DSRS owner account.

b. Define the ORACLE_SID environment variable for the DSRS V6.05 database.

```
setenv ORACLE_SID dsrsdb
```

c. Change to the directory created in Section 5.5.2.3, step c.

```
cd /dsrscm/testdb/v52tov605/upgrade
```

d. Execute the file **$DSRS_TESTDB/v52tov605/update52to605_4_sol23** which will:

(1) Check that the **/dsrscm/wais** directory exists.
(2) Check that the user has write access to the **/dsrscm/wais** directory.
(3) Check that the user has write access to the current directory.
(4) Drop the temporary database conversion tables.
(5) Create the /dsrscm/wais domain directories.
(6) Execute **/dsrscm/bin/index_keywords.sol23** for all RAs in each domain for each Key file to create the keyword database.
(7) Export the RCL tables into the update_4.exp file.

e. Check that files were successfully created in the /dsrscm/wais/domain_0 directory. Verify that the update_4.exp file exists in the current directory.

The following command will perform the above steps:

```
$DSRS_TESTDB/v52tov605/update52to605_4_sol23
```

5.6 GENERAL SYSTEM ADMINISTRATION FOR SOLARIS 2.3.

5.6.1 Tape Drives. The system configuration will have one of the three tape controllers listed in the *System Administration Manual for the DSRS*. (Also, listed in the table is the device abbreviation as it appears in the **/dev** directory, and the width of the tape it supports.)

The two tape controllers for the DSRS systems are listed in Table 5-II. (Also, listed in the table is the device abbreviation as it appears in the **/dev** directory, and the width of the tape it supports.)

Table 5-II. Sun Tape Controllers

Tape Controller	Device Abbrev	Width
Xylogics 472	rmt	1/2 inch
SCSI	rmt	1/4 inch

The status feature of the **mt (1)** command will be utilized to determine what type of tape drive will be used. For example:

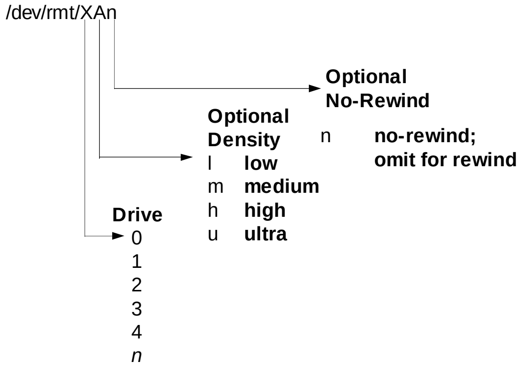
a. A tape shall be loaded in the drive on which the user will want information.

b. The command **mt -f /dev/*tape* status** will need to be entered where *tape* is rmt/0. A second or third drive, when supported, will be rmt/1 and rmt/2, respectively. Table 5-III lists the tapes on a SCSI tape controller.

Table 5-III. Tapes on a SCSI Tape Controller

Tape	Format	Tracks	Tape Length	Capacity
rmt/01	QIC-11	9	450 ft	45 Mbytes
rmt/01	QIC-11	9	600 ft	60 Mbytes
rmt/0m	QIC-24	9	450 ft	45 Mbytes
rmt/0m	QIC-24	9	600 ft	60 Mbytes
rmt/0h	QIC-150	18	600 ft	150 Mbytes
rmt/0	8mm	N/A	6000 ft	2.3 Gbyte
rmt/0	8mm	N/A	13000 ft	5 Gbytes

5.6.1.1 <u>Determining Available Tape Drives.</u> Within the /dev/rmt subdirectory is a single set of tape device files that support different output densities. In general, you specify a tape drive device as shown below.



The next three sections describe drive numbers, the optional density choices, and the optional no-rewind.

Normally, you specify a tape drive by its logical unit number, which is a number from 0 to *n*. If you do not specify a density, the drive writes at its "preferred" density, which is usually the highest density the tape supports.

To specify the first drive, use:

/dev/rmt/0

To specify the second drive, use:

/dev/rmt/1

5.6.2 __User Accounts.__ The username and the corresponding password are the most critical security barrier in the Solaris environment. The **passwd** administrative database contain user names, encrypted passwords, and other critical information. The **passwd** file contains information about every system and user account that is able to locally login to that host. It contains information about important system accounts. The **/etc/shadow** file will contain the encrypted passwords.

5.6.2.1 __The Passwd and Shadow Files.__ When users login to a Sun host, the **login** program consults the **passwd** database and verifies the user name, and consults the **/etc/shadow** file to verify the password. If the user name is not in the **passwd** database or the password is not correct for the user name, **login** denies the user access to the host. When a user name and correct password is entered, **login** grants the user access to the host.

Each entry in the passwd database has the fields as described in the table below.

Each entry in the **/etc/shadow** file has this syntax:

username::::::::

The parameters are briefly described in Table 5-IV. If the Solaris has been installed with the security option, the encrypted password will not appear here. Instead, the encrypted passwords are placed in the **/etc/shadow** file which cannot be read by ordinary user.

Table 5-IV. Parameters for passwd Database Entries

Parameter	Description
username as the *user name*	User or system account login name with two to eight characters; also referred as user id.
User ID	Account numerical user ID.
Password Status	You cannot create an account that does not require a password, and you cannot define a password from Database Manager. Use either nispasswd, yppasswd, or passwd to create a password for an account.
Comment (GCOS)	User's real name and other identifying information is printed in the user's mail message heading.
Home Path	Full pathname of the account's home directory.

Parameter	Description
Shell	Shell the account accesses upon login.
gid	Numerical ID of the group to which the account belongs.
Max Days Valid	Enter the maximum number of days the password will be valid. If you leave the field blank, the password does not expire.
Days Warning	Enter the number of days to begin warning the user befor the password expires.
Last Mod Date	The date when the password was last changed is displayed in this field as the number of days since January 1, 1970.
Expiration Date	Enter the absolute date that the user account expires, as the number of days since January 1, 1970. If you leave this field blank, the password does not expire.
Min Change Days	Enter the minimum number of days allowed between password changes.
Max Inactive Days	Enter the number of days an account can go unused (no login) before it is locked.

5.6.2.2 Setting up a Solaris Account. The following procedures show how to modify the **passwd** database which creates a user account.

a. Basic account requirements for the **passwd** file entry include: the **username** must be unique on the local node, the **gcos-field** is the full name of the user, and the login shell is **/bin/captive.** The preferred login shell cannot be the Bourne shell.

b. The user must login as a "superuser", and have OpenWindows started.

c. Start the Administration tool.

```
admintool &
```

d. Click SELECT on User Account Manager icon.

e. Click SELECT on either the NIS+ naming service or None, and click SELECT on the Load button. The User Account Manager window is displayed.

f. Choose Add User from the Edit menu. The Add User window is displayed.

g. Type the username and user ID in the appropriate text fields.

h. The default password status is to be Normal Password. This allows the password to be set during account creation.

i. Type the rest of the user account information into the text fields. Table 5-IV describes the contents of each field.

j. The home directory can be created during the account creation by selecting the create home directory block.

k. When you have entered all the information correctly, click SELECT on Add. The information is entered into the Passwd database.

l. To complete the account set-up after editing the Passwd database, you must move the necessary files into the user's directory.

```
# cp /dsrscm/config/cshrc  /home/dsrsuser/.cshrc
# cp /dsrscm/config/login  /home/dsrsuser/.login
# cp /dsrscm/config/dsrs  /home/dsrsuser/.Xdefaults
# cp /dsrscm/config/xinitrc /home/dsrsuser/.xinitrc
# cp /dsrscm/config/mwmrc /home/dsrsuser/.mwmrc
# chown user_id#  /home/dsrsuser/.cshrc
# chown user_id#  /home/dsrsuser/.login
# chown user_id#  /home/dsrsuser/.Xdefaults
# chown user_id#  /home/dsrsuser/.xinitrc
# chown user_id#  /home/dsrsuser/.mwmrc
```

m. Set the password for the newly created Solaris account, using the appropriate command depending on the systems NIS configuration.

```
(No NIS)    /usr/bin/passwd       user
(NIS)       /usr/bin/yppasswd     user
(NIS+)      /usr/bin/nispasswd    user
```

n. Set the password to expire after 60 days; with the appropriate password command for your system:

```
/usr/bin/nispasswd -x 60 user
```

o. Exit from the root shell.

5.6.3 Creating Solaris Groups.

5.6.3.1 **User Groups.** Traditional UNIX user groups consist of individuals who use the same set of files and are granted the same set of permissions. See the *System Administration Manual for the DSRS* for detailed information on adding UNIX groups. This section explains how to set up user groups on the local machine. The system administrator, logged in as **root**, will be responsible for modifying the **group** database file.

Two groups are required for the DSRS; these are **dba**, and **dsrsadm**. The **dba** group, created during installation of the ORACLE software, will allow users assigned to this group to startup and shutdown the ORACLE databases. The **dsrsadm** group will allow DSRS Librarians and Supervisors access to protected DSRS executables.

5.6.3.2 **The Group File.** In the Solaris environment, the basic form of group protection is the group database. On the local machine, this database takes the form of the **Group** database file or the **/etc/group** file.

Each entry in the **/etc/group** file has this syntax:

groupname:gid:user,user

Terminate each field with a colon, as in the **passwd** file. The **user-list** must be separated with commas and must have no spaces between user names.

The parameters are briefly described in Table 5-V.

Table 5-V. Parameters for group Database Entries (Group Access)

Parameter	Description
groupname	Name of the group.
gid	Group's numerical user ID.
user-list	List of users in the group.

5.6.3.3 **Setting Up a Solaris Group.** The following procedures show how to modify the **group** database file that creates Solaris groups. The example shown below is for the group **dba**, and **dsrsadm**, that includes all users who are librarians and supervisors of the DSRS.

a. The user must login as a "supervisor", and have Open Windows started.

b. Start the Administration tool.

```
admintool &
```

c. Click SELECT on the Database Manager icon. The Database Manager window is displayed.

d. Click SELECT on the Group database, then click SELECT on either the NIS+ naming service or None.

e. Click SELECT on Load. The Group Database window is displayed.

f. Choose Add Entry from the Edit Menu. The Group Database: Add Entry window is displayed.

g. Type the group name in the appropriate text field.

h. Type the group ID in the appropriate test field.

i. Type the list of members in the appropriate text field, separated by a comma.

j. Click SELECT on Add. The group is added to the Group database.

k. Exit from the root shell.

5.6.3.4 **Assigning Users to Groups.** The last field for each entry in the **group** database file includes all the Solaris account login names of all users who belong to that group. All DSRS Supervisors and Librarians must belong to the **dsrsadm** group. All Solaris users who need privilege to startup and shutdown ORACLE databases must belong to the **dba** group.

The procedures below show how to modify the **group** database file to add or remove users.

a. The user must login as a "supervisor", and have Open Windows started.

b. Start the Administration tool.

```
admintool &
```

c. Click SELECT on the Database Manager icon. The Database Manager window is displayed.

d. Click SELECT on the Group database, then click SELECT on either the NIS+ naming service or None.

e. Click SELECT on the name of the group you want to modify. The Group Database window is displayed.

f. Choose Modify Entry from the Edit Menu. The Group Database: Modify Entry window is displayed, showing the current entries for the group.

g. Add or delete user names from the text field, separated by a comma.

h. Click SELECT on Modify. The entry in the Group database is modified.

i. Exit from the root shell.

5.6.4 <u>ORACLE Database Startup.</u> ORACLE provides a shell script for performing a database startup during a system boot procedure, and ORACLE also provides an SQL*DBA tool to interactively startup a database.

5.6.4.1 <u>Interactive Method.</u> The SQL*DBA is a tool for helping database administrators manage and monitor a database. SQL*DBA may be used interactively to startup a database. Refer to the *ORACLE7 Server Administrator's Guide* for detailed information on the SQL*DBA tool.

The example below defines the steps necessary for performing an interactive database startup. The steps will work only for those SunOS users who are members of the **dba** group.

a. Define the necessary ORACLE logicals to point to the database which you will be starting:

```
setenv ORACLE_SID dsrsdb
setenv ORACLE_HOME /usr/oracle7
set path = ($PATH $ORACLE_HOME/bin)
```

b. Connect to the SQL*DBA tool:

```
sqldba
```

c. At the **SQLDBA** input screen, enter the following commands using the database name of the database defined by the ORACLE_SID:

```
connect internal
startup open dsrsdb
```

NOTE: The DSRS installation created a database whose **ORACLE_SID** and database name were the same.

d. Exit from the SQL*DBA tool with the command:

```
exit
```

5.6.4.2 <u>DBSTART Method.</u> ORACLE provides a shell script to ensure a clean, automatic database startup. To ensure that the ORACLE database will come up after the Solaris has been shutdown and restarted, the dbstart shell script can be linked to files in **/etc/rc2.d** directory. This script will bring up all the databases listed in **/var/opt/oracle/oratab** file that have the third field specified as **Y** (yes). In addition, this script will startup the orasrv process and the msql daemon.

Create a file named **dbora** in the **/etc/initd** directory.

Confirm that there are entries like the following at the end of the file (be sure to give the full path of the **dbstart** utility):

```
ORACLE_HOME=/opt/oracle
ORA_HOME=/opt/oracle
ORA_OWNER=oracle
su - $ORA_OWNER -c $ORA_HOME/bin/dbstart &
su - $ORA_OWNER -c $ORA_HOME/bin/orasrv
/usr/local/Minerva/bin/msqld &
```

If these entries do not exist, you need to add them to the file.

Link this file to files in **/etc/rc2.d** by entering the following:

```
# ln -s /etc/init.d/dbora  /etc/rc2.d/K98dbora

# ln -s /etc/init.d/dbora  /etc/rc2.d/S98dbora
```

5.6.5 **ORACLE Database Shutdown.** ORACLE does not provide a shell script for performing a database shutdown during a system shutdown procedure. However, ORACLE does provide a shell script, dbshut, to shutdown all of the databases listed in the **/etc/oratab** file that have the third field specified as **Y** (yes); ORACLE also provides an SQL*DBA tool to interactively shutdown a database.

5.6.5.1 **Interactive Method.** The SQL*DBA is a tool for helping database administrators manage and monitor a database. SQL*DBA may be used interactively to shutdown a database. Refer to the *ORACLE7 Server Administrator's Guide* for detailed information on the SQL*DBA tool.

The example below defines the steps necessary for performing an interactive database shutdown. The steps will work only for those SunOS users who are members of the **dba** group.

a. Define the necessary ORACLE logicals to point to the database which you will be shutting down:

```
setenv ORACLE_SID dsrsdb
setenv ORACLE_HOME /usr/oracle7
set path = ($PATH $ORACLE_HOME/bin)
```

b. Connect to the SQL*DBA tool:

```
sqldba
```

c. At the **SQLDBA** input screen, enter the commands below:

```
connect internal
shutdown abort
```

d. Exit from the SQL*DBA tool with the command:

```
exit
```

5.6.5.2 **DBSHUT Method.** ORACLE provides a shell script to shutdown all databases. The **dbshut** script will shutdown all of the databases listed in the **/etc/oratab** file that have the third field specified as Y (yes).

The **dbshut** script can only be run interactively. The following line will shutdown the ORACLE databases when executed by the superuser:

```
su - oracle -c /usr/oracle7/bin/dbshut
```

5.7 DATABASE/DATA BANK.

5.7.1 **Database Files.** ORACLE RDBMS uses several features of SunOS to provide a secure environment for the user community. The SunOS features include file ownership, group accounts, and the ability to have a program change its user ID upon execution. The directories containing the database files and indexes should be owned by the ORACLE user ID. Group and world users should not have write access to the database. Table 4-II indicates all permanent database files which will be used by the system.

a. The **Database** row identifies the database instance used by the DSRS.

b. The **Tablespace** row identifies the names of the four tablespaces contained in the DSRS database. The SYSTEM tablespace exists in every ORACLE database and is created automatically during installation. The SYSTEM tablespace contains the data dictionary for the entire DSRS database, with names and locations of all database objects (such as tables, indexes, and other tablespaces). The other three tablespaces are unique to the DSRS.

c. The **Files** row identifies the name of the database file. The database filename is defined during execution of the **/dsrscm/testdb/dsrs_install** as the first variable defined in the input script.

d. The **Directory** row identifies the directory structure where the database files and index files are located. The logical $DB_LOC is defined during execution of the **/dsrscm/testdb/dsrs_install** as the second variable defined in the input script, and the logical $INDEX_LOC is defined as the third variable defined in the input script.

e. The **Size** row identifies the size of the database file in bytes.

A qualified database administrator can increase the size of the tablespaces using the SQL*Plus command:

```
alter tablespace tablespace_name add datafile file_name size 30M.
```

5.7.2 **ORACLE Process Architecture.** An ORACLE database is comprised of four background processes and the System Global Area (SGA). The database is known as an instance, and only one instance may point to one database at any given time. The four background processes are listed below; detailed information can be found in the *ORACLE7 Server Administrator's Guide.*

a. **LGWR - Redo Log Writer**

b. **DBWR - Database Writer**

c. **PMON - Process Monitor**

d. **SMON - System Monitor**

5.7.3 ORACLE Storage Requirements. Refer to Table 2-III of the *System Administration Manual for the DSRS* which lists the storage requirements for the ORACLE tools used by the DSRS.

5.8 OUTPUT REPORTS. The DSRS does not generate any reports during installation; however, the SunOS has an error logging daemon, **syslogd**, that will be used by many system facilities to record error messages whenever an unusual event occurs. These messages will be written to the file **/var/adm/messages**. This file will be available for viewing errors that occur on the system.

ORACLE errors, due to the failure of one of the background processes, will be recorded in trace (dump) files generated by the process. These files will be found in the location specified by the INIT.ORA parameter, BACKGROUND_DUMP_DEST, and will give some information on why the process may be failing.

5.9 DIAGNOSTIC MESSAGES. If the tape media cannot be read, the operating system will respond with error messages. An error may be caused by one of these conditions: the tape is bad, the reading was interrupted by a power failure, or some other reason. If the tape media is bad, contact the SRP. If the installation procedure failed, rewind the tape and begin the installation procedure again.

5.10 RESTART/RECOVERY PROCEDURES. The following subsections give an overview of items to check when debugging problems are encountered during a COTS or a DSRS installation.

5.10.1 Basic Troubleshooting. Table 5-VI lists some common errors that may be encountered during a COTS or DSRS installation.

Table 5-VI. Common Errors During Installation

ERROR	EXPLANATION
tar: /dev/rst8: No such device or address	The incorrect tape drive device was entered. Refer to Section 5.6.1 of this document for determining tape drive devices.
tar: /dev/rst8: I/O error	The tape was not inserted into the tape drive.

ERROR	EXPLANATION
Usage: dsrs_install new_sid db_file_loc	The new unique ORACLE_SID and the pathname of the location where the new database files are to be installed.
Sorry	An incorrect account password was entered.
Unknown login: account name	The user account name was entered incorrectly.
No directory! Logging in with home=/	The account's home directory pathname was not created. Refer to Section 5.3.1 of this document for steps to create a home directory pathname.
ORA-01031: insufficient privileges	The account executing the SQL*DBA commands is not a member of the **dba** group. Refer to Section 5.6.3 of this document for steps to add an account to a group.
Xlib: Connection to ":0.0" refused by server *Error: Can't open display*	An xterm session was unsuccessful because the user process starting the xterm window is not the same user process which started the OpenWindows session.

5.10.2 **Debugging DSRS Problems.** If problems are encountered when using the DSRS, the following steps shall be taken to determine the problem and its resolution:

a. Use the **env** command to verify that the DSRS environment variables have been defined. If the variables have not been defined, refer to the *System Administration Manual for the DSRS* for instructions to define them.

b. The four database processes that are currently active on the system will need to be verified. Refer to Section 5.10.3 of this document for instructions to determine what processes are available, what processes are not currently active, and how to bring them up with the **dbstart** command.

c. Verify that the DSRS database tables have data stored in them. If the tables are empty or corrupt, the database tables will need to be restored using the ORACLE import tool described in the *System Administration Manual for the DSRS*.

d. Although performing a system reboot is a regular procedure, if the above situations are not causing the problems, then rebooting the system may fix any problems with the DSRS. Refer to the *System Administration Manual for the DSRS* for steps to shut down and start up the operating system.

e. If the DSRS is still not accessible to users, then contact the SRP for further help.

5.10.3 <u>Debugging ORACLE Problems.</u> The following items will need to be checked to make sure that the ORACLE database is running properly. If an ORACLE error message is received at anytime, refer to the *ORACLE7 Server Messages and Codes Manual* for error resolution procedures.

Verify that ORACLE has been installed according to the *ORACLE7 for Sun SPARC Solaris 2.3 Installation and User's Guide*.

a. It will need to be verified (described in Section 5.7.2 of this document) that the four ORACLE processes are currently executing on the system. This can be done with the following command:

```
ps -ef
```

b. If all four processes are executing, then continue to paragraph d. If all four processes are not executing, then execute an interactive ORACLE shutdown and startup as described in Sections 5.6.5.1 and 5.6.4.1 of this document respectively.

c. If the database and its four processes do not come up, then refer to the *ORACLE7 Server Messages and Codes Manual* for resolution of the specific error message received when performing the interactive shutdown or startup.

d. If the database is up and problems still occur, verify that the tablespaces (described in Section 5.7.1) are available and that the DSRS database tables have data stored in them.

e. If the database tablespaces or the data in the tables are not available, then restore the database information from an ORACLE export file. The steps for restoring database information are detailed in the *System Administration Manual for the DSRS*.

5.10.4 <u>Debugging SunOS Problems.</u> Typically, a process shall never crash the host system. However, a user process may "hang" the system (a user process, a user's window, or even the whole system) even though the system will appear to be running.

Sometimes the system may appear as hung, that is, it will not respond to anything typed. Before assuming that the program has crashed, check the items below:

a. Type **CTRL q** in case the screen was frozen by a **CTRL s** keystroke.

b. The **tty** mode may be corrupted. The line feed character may be typed, **CTRL j**, instead of the **RETURN** key, which will force a line feed. If the system responds, **CTRL j /usr/ucb/reset CTRL j** will need to be typed to reset the **tty** modes.

c. If the window system is running, make sure the cursor resides in the active window where the user is trying to type commands.

d. Type **CTRL \**. This shall force a "quit" in the running program, and probably the writing of a core file.

e. Type **CTRL c** to interrupt the program that may be running.

f. If possible, try logging into the same CPU from another terminal, or remote login from another system on the network. Type **ps -ef**, and look for the hung process. If hung processes are identified, try to kill the hung processes. To kill a process, the user must be logged in as the same user running the process or logged into the root account. Type **kill <pid number>**. If this does not work, try **kill -9 <pid number>**. (A quick way to see if **kill** has worked is to repeat it. If the response is **no such process**, it was killed.)

g. If all of the above fail, abort and reboot the host system.

h. If it continues to fail, call Sun Customer Service for help.